REMARKS

This amendment is responsive to the Office action mailed December 3, 2003 for the above-captioned application.

- Claims 1-4, 13-17 and 22 have been rejected under 35 USC 102(e).
- Claims 5-6 and 18-21 have been rejected under 35 USC 103(a).
- Claims 1-12 and 14-22 are canceled.
- Claim 13 is amended.
- New claims 23-25 are added.
- Claims 13 and 23-25 are in independent format
- Claims 13 and 23-25 are pending.

The Cited Art

The examiner has rejected claim 13 under 35 USC 102(e) as being anticipated by U.S. Patent No. 6,401,190 (Nishioka et al.). Nishioka et al. disclose a processor having parallel processing units. Each processing unit 22-25 includes two computing units 26, 27 and a register file 28. An instruction includes multiple fields. A Nishioka et al. processing unit is compared to applicants' cluster. A Nishioka et al. computing unit (e.g., IFG or INT) is compared to applicants' functional processing unit. A Nishioka et al. field is compared to applicants' subinstruction. Nishioka et al. disclose 4 processing units 22-25 with two computing units 26, 27 (IFG, INT) within each processing unit. A instruction can have up to 8 fields.

Nishioka et al. describe a method for expanding an instruction. In some embodiments the expansion occurs without copying a field. Nishioka et al. parse a header for a given instruction to determine which fields are expressly listed in the unexpanded instruction (see Fig. 4) of Nishioka et al. The header includes 8 bits for an instruction. A'0' in a given bit position designates a no-op operand for a corresponding field in an expanded instruction. A '1' in a given bit position designates that a field follows for the corresponding field in the expanded instruction.

Of relevance here is the embodiment where a field is copied to other fields during expansion of the compressed instruction (col. 2, lines 59-67). The examiner's attention is directed to the SIMD mode at col. 13, line 25 through col. 14, line 56, and especially to col. 14, lines 38 to 56; and to Figs. 1, 2 and 6.

Referring to Nishioka Fig.2, the field format for an IFG field and for an INT field are shown. When the s-mode bit is set to '1' in the IFG field, then SIMD mode is active for that IFG field and a paired INT field. In such case the three SIMD bits 9-11 designate which of the other processing units 22-25 are to receive a copy of the IFG and INT fields. Note that copying occurs only in pairs, as shown in Fig. 6. For further detail on this limitation, see also the dual selector 48 of the instruction expansion circuit (Fig. 7), the SIMD controller of Figs. 7 and 16, and especially the description of the SIMD controller at col. 27, lines 38-49. Thus, Nishioka et al.'s method requires that both fields (i.e., IFG, INT) of a processing unit be copied when copying occurs.

It is noted that Nishioka et al. do not disclose or suggest a manner of copying a field of one computing unit (e.g., IFG) to the field of another non-corresponding computing unit (e.g., INT) within a different processing unit. The corresponding units to the IFG in one processing unit are the IFGs in the other processing units. (Compare applicants' claim 24).

It also is noted that Nishioka et al. use 4 control bits to control field copying of a given field pair. Referring to Nishioka Fig. 2, bit 27 determines whether SIMD mode is on, and bits 9-11 determine which of the processing units is to receive a copy. This limits the number of different copy permutations omitting many potential permutations. Compare applicants' claims 13 and 23 where all possible permutations of copying are possible.

It is also noted that Nishioka et al. disclose four copy control bits (SIMD bits) for every field of the uncompressed instruction where SIMD mode is active. Thus, the aggregate number of copy control bits for a compressed instruction is some multiple of four, where the specific number may vary from one compressed instruction to the next. Compare this to applicants' claim 25 where there is a fixed number of control bits for the compressed instruction regardless of how many subinstructions are expressly included.

For comparing terminology it is noted that Nishioka et al. include a fixed number of copy control bits in a field (a field instruction), resulting a variable number of control bits in the compressed VLIW instruction due to the varying number of field instructions in the compressed VLIW instruction. In contrast applicants do not disclose any copy control bits within a subinstruction format. Applicants disclose control bits in the VLIW instruction, and more specifically disclose a fixed number of control bits for the compressed VLIW instruction regardless of the number of subinstructions in the compressed VLIW instruction.

Claims 13 and 23:

Claim 13 distinguishes over the cited art based at least upon the following claim limitations:

- each instruction to be executed by the processor comprising from one subinstruction up to the second prescribed number of subinstructions, and a set of control bits, the set of control bits including a first subset of control bits equal in number to the second prescribed number, wherein the first subset of control bits identify all expressly included subinstructions which are to be shared and a routing pattern for distributing all the shared subinstructions; and
- the at least one subinstruction to be shared by a plurality of the functional processing units, said plurality of functional processing units being determined by said condition of the first subset of control bits, wherein the first subset of control bits allows as many as 2^z permutations for subinstruction sharing within any given VLIW instruction, where z equals said second prescribed number. (The second prescribed number equals the number of clusters times the common number of functional units per cluster).

By allowing up to 2^z permutations, applicants are able to achieve every possible sharing permutation among all the functional processing units. Nishioka et al. have a limited capability of field copying and do not disclose or suggest any means for achieving every possible permutations.

Claim 23 includes similar limitations as claim 13 regarding the subset of control bits and the to 2^z subinstruction sharing permutations. Accordingly claim 23 distinguishes over the cited art based on similar reasons as given for claim 13.

<u>Claim 24:</u>

Claim 24 distinguishes over the cited art based at least upon the following claim limitations:

- wherein each one i^{ith} functional processing unit in a given cluster has one corresponding functional processing unit and at least one non-corresponding functional processing units in the other clusters of the plurality of clusters;
- testing the set of control bits;
- wherein for a first prescribed condition of the set of control bits, at least one expressly included subinstruction is routed to non-corresponding functional processing units as determined by the first prescribed condition.

Nishioka et al do not disclose or suggest copying a field for one computing unit (e.g., IFG) of one processing unit to a non-corresponding computing unit (e.g., INT) of a different processing unit.

<u>Claim 25:</u>

Claim 25 distinguishes over the cited art based at least upon the following claim limitations:

- a processor having a very large word instruction architecture and including a plurality of clusters of functional processing units, each one cluster of the plurality of clusters comprising a common number of functional processing units, the processor comprising a first prescribed number of clusters; and
- a memory which stores a VLIW instruction, wherein <u>the VLIW instruction includes a set of control bits and up to a second prescribed number of subinstructions, where the</u>

second prescribed number equals the first prescribed number times the common number,

- wherein the set of control bits comprises a fixed length subset of control bits which identify a routing pattern for distributing the up to the second prescribed number of subinstructions, including a designation for sharing at least one subinstruction, wherein said fixed length is the same independently of the number of subinstructions included in the VLIW instruction.

Nishioka et al. do not provide a set of control bits for controlling field copying where the set of control bits encompasses the entire VLIW instruction. In Nishioka the control bits are for a pair of fields, not for all fields. Further in Nishioka the number of control bits varies according to the number of fields expressly included in the compressed VLIw instruction and is not fixed as required by claim 25. The method of claimed 25 allows for a simpler control scheme which can achieve fewer or more permutations according to a design choice for implementing the fixed number of control bits.

Conclusion

In view of the above remarks regarding the cited art, it is respectfully submitted that the claims contain key limitations that are not present in the cited art and not obvious from the cited art. These particular limitations, are not disclosed in or suggested by cited references. These limitations are significant advances over the prior art and resulted in a novel method and apparatus for sharing VLIW subinstructions.

In view of the above amendments and remarks, it is respectfully submitted that the claims are now in condition for allowance. The Examiner's action to that end is respectfully requested. Reconsideration of the claims and withdrawal of the rejections is respectfully requested.

If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the application, the Examiner is invited to call the undersigned attorney at the telephone number given below.

Respectfully submitted,

Dated: 3/1/2004

By: P. Koda
Steven P. Koda
Reg. No. 32,252
Tel.: 360-859-4013

Koda Law Office
8070 E. Mill Plain Blvd, No. 141
Vancouver, WA 98664-2002